

Study Guide: Salesforce Fundamentals (8%)

Overview

The Salesforce Fundamentals section of the Salesforce Platform App Builder Certification Exam, weighted at 8%, tests your core understanding of the Salesforce platform, its architecture, and its customization capabilities. This topic lays the groundwork for all other exam sections, ensuring you grasp what Salesforce is, how it functions as a Customer Relationship Management (CRM) system, and how to extend it using declarative tools and AppExchange. While it's a smaller portion of the exam, it's critical for contextualizing the more technical topics like Data Modeling or Process Automation.

You'll need to master the platform's purpose, standard objects, declarative vs. programmatic customization, multitenancy, and the role of AppExchange. This guide will dive deep into each area, offering practical insights, real-world scenarios, and study aids to help you succeed on the exam and in real-world Salesforce implementations.

Key Concepts

Here's what you'll need to master:

1. **Salesforce Platform Overview:** Purpose, CRM functionality, and cloud architecture.
 2. **Standard Objects:** Core building blocks like Accounts, Contacts, and Opportunities.
 3. **Declarative Customization:** Point-and-click tools for building apps.
 4. **Programmatic Customization:** When coding (Apex, Lightning Components) is needed.
 5. **Multitenancy:** Shared infrastructure with isolated data.
 6. **AppExchange:** Marketplace for prebuilt solutions.
 7. **Salesforce Editions:** Differences in features and limits.
 8. **Use Cases:** Applying fundamentals to business needs.
-

Detailed Explanation

1. Salesforce Platform Overview

Salesforce is a cloud-based CRM platform designed to manage customer relationships, streamline business processes, and drive growth. It's built on a multitenant architecture and offers extensive customization options.

- **Purpose:**
 - Centralize customer data (e.g., contacts, sales, support).
 - Automate workflows (e.g., lead follow-ups).
 - Provide analytics (e.g., sales forecasts).
- **Cloud Architecture:**
 - **Software as a Service (SaaS):** No hardware management—access via browser.
 - **Platform as a Service (PaaS):** Build custom apps on the Lightning Platform.
- **Key Features:**
 - CRM: Sales, Service, Marketing modules.
 - Customization: Declarative and programmatic tools.
 - Scalability: Grows with business needs.
- **Example:** A retailer uses Salesforce:
 - Tracks customers (Accounts), sales (Opportunities), and support (Cases).
- **Real-World Scenario:** A nonprofit manages donors:
 - Uses Salesforce to log donations, automate thank-you emails, and report on fundraising.
- **Best Practices:**
 - Start with business goals—map Salesforce to processes.
 - Leverage out-of-the-box features before customizing.
 - Understand cloud benefits (e.g., automatic upgrades).

2. Standard Objects

Standard Objects are prebuilt data containers in Salesforce, forming the core of its CRM functionality.

- **Key Objects:**
 - **Account:** Companies or organizations (e.g., “Acme Corp”).
 - **Contact:** Individuals tied to Accounts (e.g., “John Doe, Acme Corp”).
 - **Opportunity:** Potential sales deals (e.g., “Acme Q3 Deal”).
 - **Lead:** Prospects not yet qualified (e.g., “Jane Smith, Trade Show”).
 - **Case:** Support tickets (e.g., “Acme Billing Issue”).
- **Relationships:**
 - Account → Contact (Master-Detail or Lookup).
 - Opportunity → Account (Lookup).
 - Lead → Converted to Account/Contact/Opportunity.
- **Example:** Sales process:
 - Lead → Qualify → Convert to Contact + Opportunity → Close deal.
- **Real-World Scenario:** Service team:
 - Case linked to Contact and Account → Track issue resolution.
- **Tips:**
 - Explore standard fields (e.g., StageName on Opportunity).
 - Use standard objects when possible—reduces custom work.
 - Understand object limits (e.g., 25 Lookup relationships per object).

3. Declarative Customization

Declarative Customization uses point-and-click tools to tailor Salesforce without coding, ideal for App Builders.

- **Tools:**
 - **Object Manager:** Create custom objects, fields, relationships.
 - **Lightning App Builder:** Design UI pages.
 - **Flow:** Automate processes.
 - **Setup:** Configure security, layouts, etc.

- **Example:** Add a custom field:
 - Object: Opportunity → Field: Discount__c (Percent).
- **Real-World Scenario:** A company needs project tracking:
 - Custom Object: Project__c → Fields: Start_Date__c, Budget__c.
 - Flow: Auto-notify manager when budget exceeds threshold.
- **Best Practices:**
 - Prioritize declarative over code—faster and maintainable.
 - Document customizations for future admins.
 - Test in a sandbox first.

4. Programmatic Customization

Programmatic Customization involves coding when declarative tools can't meet complex requirements.

- **Tools:**
 - **Apex:** Salesforce's programming language (like Java).
 - **Visualforce:** Custom UI pages.
 - **Lightning Components:** Modern UI framework (Aura or Lightning Web Components).
- **When to Use:**
 - Complex logic (e.g., custom calculations across objects).
 - Unique UI (e.g., interactive dashboard).
- **Example:** Custom Opportunity scoring:
 - Apex trigger to calculate Score__c based on multiple fields.
- **Real-World Scenario:** A retailer needs a custom checkout:
 - Lightning Component → Interactive form → Apex to process payment.
- **Tips:**
 - Know when to escalate to developers (exam may test this).
 - Understand limits (e.g., 100 SOQL queries per Apex transaction).

- App Builders focus on declarative—coding is supplementary.

5. Multitenancy

Multitenancy is Salesforce's architecture where all customers share the same infrastructure but keep data isolated.

- **How It Works:**
 - One codebase, one database → Separate orgs via metadata.
 - Updates applied globally (e.g., 3 releases per year: Spring, Summer, Winter).
- **Benefits:**
 - Cost-effective: No server management.
 - Scalable: Handles small startups to enterprises.
 - Secure: Data isolation via org IDs.
- **Example:** Two companies on Salesforce:
 - Company A's Accounts never mix with Company B's.
- **Real-World Scenario:** A global firm:
 - Uses one Salesforce instance → Multiple divisions with isolated data.
- **Best Practices:**
 - Understand release impacts (e.g., new features, deprecations).
 - Leverage multitenancy for cost savings.
 - Trust Salesforce security—don't over-customize.

6. AppExchange

AppExchange is Salesforce's marketplace for prebuilt apps, components, and integrations.

- **Offerings:**
 - **Apps:** Full solutions (e.g., billing software).
 - **Components:** UI add-ons (e.g., enhanced calendars).
 - **Consulting:** Partner services.
- **Types:**

- Managed: Locked, upgradeable packages.
- Unmanaged: Open, editable packages.
- **Example:** Install a reporting tool:
 - App: “Advanced Analytics” → Adds custom dashboards.
- **Real-World Scenario:** A company needs e-signatures:
 - AppExchange: Install DocuSign → Integrates with Opportunities.
- **Tips:**
 - Evaluate apps for fit (ratings, reviews, trial).
 - Check compatibility with your edition.
 - Use before building custom—saves time.

7. Salesforce Editions

Salesforce offers editions with varying features and limits, impacting what you can build.

- **Editions:**
 - **Essentials:** Small businesses (basic CRM, 10 users max).
 - **Professional:** Growing businesses (limited customization).
 - **Enterprise:** Large orgs (advanced features, API access).
 - **Unlimited:** Maximum flexibility (unlimited customizations).
- **Key Differences:**
 - Custom Objects: Essentials (10), Enterprise (200).
 - API: Not in Professional, included in Enterprise+.
- **Example:** A startup:
 - Essentials → Basic Leads, Accounts, no custom apps.
- **Real-World Scenario:** A corporation:
 - Enterprise → Custom Order__c, Flow automation.
- **Tips:**

- Know edition limits for exam scenarios (e.g., “Can X be done in Professional?”).
- Enterprise is most common—focus there.
- Check Salesforce documentation for current limits.

8. Use Cases

Applying fundamentals to business needs ties everything together.

- **Sales:** Track leads → Opportunities → Closed deals.
- **Service:** Manage Cases → Resolve issues → Report SLA.
- **Marketing:** Capture Leads → Run campaigns → Measure ROI.
- **Example:** Small business sales:
 - Standard Objects: Leads, Accounts, Opportunities.
 - Declarative: Custom field Lead_Source__c.
- **Real-World Scenario:** Nonprofit donor management:
 - Custom Object: Donation__c.
 - AppExchange: Install payment processor.
- **Best Practices:**
 - Map use cases to objects/tools.
 - Start simple, scale as needed.
 - Validate with stakeholders.

Study Guide Tables

Table 1: Standard Objects

Object	Purpose	Key Fields
Account	Companies	Name, Industry
Contact	Individuals	FirstName, Email

Object	Purpose	Key Fields
Opportunity	Sales deals	Amount, StageName
Lead	Prospects	Company, Status
Case	Support tickets	CaseNumber, Status

Table 2: Declarative vs. Programmatic

Type	Tool	Use Case
Declarative	Lightning App Builder	Custom page layout
Programmatic	Apex	Complex data processing
Declarative	Flow	Multi-step automation
Programmatic	Lightning Component	Custom UI widget

Table 3: Editions Comparison

Edition	Custom Objects	API Access	Use Case
Essentials	10	No	Small biz CRM
Professional	25	No	Mid-size sales
Enterprise	200	Yes	Large org apps
Unlimited	2000	Yes	Enterprise scale

Practical Examples

1. **Standard Objects:** Sales process:
 - Lead → Convert to Contact + Opportunity → Close.
2. **Declarative:** Add a field:
 - Account → Customer_Tier__c (Picklist: Gold, Silver).
3. **AppExchange:** Install a tool:
 - “Chatter Plus” → Enhances collaboration.

4. **Use Case:** Service tracking:

- Case → Custom field Resolution_Time__c.
-

Tips for Success

- **Hands-On Practice:** In a Developer org:
 - Explore Accounts, Contacts, Opportunities.
 - Add a custom field declaratively.
 - Install an AppExchange app.
 - **Context is Key:** Understand why Salesforce exists (CRM + customization).
 - **Trailhead Modules:**
 - “Salesforce Platform Basics”
 - “CRM for Lightning Experience”
 - **Scenarios:** Practice questions like:
 - “Should X use declarative or AppExchange?”
 - “Which object for Y process?”
 - **Editions:** Memorize key limits (e.g., custom objects).
 - **Multitenancy:** Know release cycle (3x/year).
 - **Start Simple:** Focus on standard features first.
-

Bullet Point Summary

- **Platform Overview:**
 - Cloud CRM.
 - SaaS + PaaS.
 - Scales with needs.
- **Standard Objects:**
 - Core CRM blocks.

- Prebuilt relationships.
 - Use before custom.
 - **Declarative:**
 - Point-and-click.
 - Fast and flexible.
 - App Builder focus.
 - **Programmatic:**
 - Code for complexity.
 - Apex, Visualforce.
 - Know limits.
 - **Multitenancy:**
 - Shared infrastructure.
 - Isolated data.
 - Auto-updated.
 - **AppExchange:**
 - Prebuilt solutions.
 - Managed/Unmanaged.
 - Evaluate fit.
-

Study Plan

1. **Day 1:** Study Platform Overview and Objects (explore in org).
2. **Day 2:** Master Declarative Customization (add a field).
3. **Day 3:** Learn Programmatic and Multitenancy (review docs).
4. **Day 4:** Explore AppExchange and Editions (install an app).
5. **Day 5:** Review all concepts, test a scenario.
6. **Day 6:** Take a practice quiz on Fundamentals.

Exam Scenarios to Practice

1. **Scenario:** “A company needs basic CRM—edition?”
 - Solution: Essentials → Leads, Accounts, Contacts.
2. **Scenario:** “Track custom projects—declarative or code?”
 - Solution: Declarative → Project__c object.
3. **Scenario:** “Add e-signatures—how?”
 - Solution: AppExchange → Install DocuSign.